UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/672,508	09/26/2003	Paul W. Coleman	MSFT4	3534

27488 7590 10/18/2006

MERCHANT & GOULD (MICROSOFT)
P.O. BOX 2903
MINNEAPOLIS, MN 55402-0903

EXAMINER	
RUTLEDGE, AMELIA L	

ART UNIT	PAPER NUMBER
2176	

DATE MAILED: 10/18/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/672,508	COLEMAN ET AL.	
	Examiner	Art Unit	
	Amelia Rutledge	2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 January 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to communications: Amendment, filed 08/04/06; Request for Continued Examination, filed 08/04/06.

2. Claims 1-22 are pending in the case. Claims 1, 5, 9, and 13 are independent claims.

Continued Examination Under 37 CFR 1.114

3. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 08/04/06 has been entered.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-4 and 15-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over "CAST Bobby: Manual for Downloadable Version 2000" (hereinafter "Bobby"), copyright 1996-2000 Center for Applied Special

Technology (CAST), in view of Duggan et al. (hereinafter "Duggan"), U.S. Patent No. 6,002,871, issued December 1999, and further in view of Kondoh et al. (hereinafter "Kondoh"), U.S. Patent No. 6,886,115 B2 issued April 2005.

Independent claim 1 cites: *A computer-executable method of testing a hypertext document for compliance with a selected criterion, said method comprising: accepting a user selection of the selected criterion and user input of at least one parameter indicative of compliance with the selected criterion for an analysis of the hypertext document*;

Bobby teaches a method of testing a hypertext document for compliance using a *bobbycl* function, which allows the user to input parameters to test for compliance to a selected criterion such as browser or HTML specification, or accessibility requirement (p. 12-13, "Using *bobbycl*"). Bobby teaches a dialog which allows a user to select the a specific browser compatibility test (p. 11, "Browser compatibility tests"). While Bobby allows a user to select from a group of compliance tests, i.e., the selected criterion, Bobby does not explicitly teach user input of a parameter indicative of compliance. However, Duggan teaches a test tool for testing web applications (Col. 1, l. 15-21) which provides a graphical user interface for user input of test parameters, i.e., a parameter indicative of compliance with the criterion (Fig. 1 and 7, Col. 4, l. 40-45). Both Bobby and Duggan are analogous art because both are testing tools for web applications. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Duggan to Bobby, so that Bobby would have the benefit of a simple, easy way to generate and modify a number of test scripts, and verify proper

execution of user functions, in a manner that required little programming expertise (Duggan, Col. 2, l. 31-46).

Claim 1 also cites: *performing the analysis of the hypertext document and generating analysis data, the analysis data comprising an indication of whether the hypertext document is in compliance with the selected criterion; correcting a definable error by presenting the analysis data to an application program; storing the analysis data and the link information for the hypertext document; and presenting at least a portion of the analysis data to the user.*

Bobby teaches performing an analysis of a hypertext document for compliance and the generation of a summary report of analysis data showing whether a document is in compliance, and storing and presenting the reports (p. 8, "View a report", p. 7-8, "Reading the report). Bobby teaches storing the analysis data and link information for the document (p. 9, Reanalyzing pages that have changed). While Bobby in view of Duggan does not explicitly teach correcting a definable error by presenting the analysis data to an application program, Kondoh teaches a method of testing a hypertext document for compliance with a selected criterion, where the user can select the criterion (col. 3, l. 50-col. 4, l. 15). Kondoh teaches correcting a definable error by presenting the analysis data to a program module, i.e., application program (col. 5, l. 13-34). Bobby, Duggan, and Kondoh are analogous art since all three inventions are directed toward error detection and parsing the structure of web applications. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the selective automatic and user defined error correction of Kondoh with the

testing GUI of Duggan and the user defined compliance testing method of Bobby, so that the result of the error correction would reflect the intent of the document's creator, thereby allowing more flexible and selective error correction (Kondoh, col. 3, l. 50-62).

Regarding dependent claim 2, Bobby discloses that the user can type in the URL of the page they want to analyze (p. 8, "Entering a URL to test"), compare to *accepting a user designation of the hypertext document to be analyzed.*

Regarding dependent claim 3, Bobby teaches that the user can limit the depth of link levels by a range of zero to Infinite levels (p. 8-9, "Choosing the scope of site analysis"). Compare to *accepting a user designation of the number of link levels from the hypertext document to be analyzed.*

Regarding dependent claim 4, Bobby discloses a *linkfinder* function which *specifies* which hypertext links to follow and which documents to analyze (p. 12, "Using *linkfinder* "), and a *bobbycl* function, which allows the user to input parameters to test for compliance to a selected criterion such as browser or HTML specification, or accessibility (p. 12-13, "Using *bobbycl*"). These functions can be run together by the user to generate an analysis of hypertext documents for compliance, and where the analysis data is stored for each document (p. 13-14, "Putting them together"). Bobby teaches presenting compiled summary information about the collection of linked hypertext documents, and identifying what criterion is most problematic, in the form of a list of all issues found, grouped by priority (p. 10, "Summary Reports").

Regarding newly added dependent claim 15, while Bobby does not explicitly teach copying the hypertext documents, Duggan teaches copying the hypertext

documents thereby providing a snapshot of each hypertext document at a point in time of analysis, because Duggan teaches displaying the HTTP response and HTML source code of the document to the user (Col. 12, l. 5-15; Col. 13, l. 1-21; Fig. 6). While Bobby in view of Duggan does not explicitly teach storing copies of the hypertext documents, it would have been obvious to one of ordinary skill in the art at the time of the invention to store copies of the hypertext documents, since storing copies of hypertext documents at a point in time for later reference was a common and notoriously well known practice in the art at the time of the invention. Further, Bobby discloses analyzing stored copies of hypertext documents on a user's computer (Bobby, p. 8, "Entering a URL to test", par. 2). Both Bobby and Duggan are analogous art because both are testing tools for web applications. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Duggan to Bobby, so that Bobby would have the benefit of a simple, easy way to generate and modify a number of test scripts, and verify proper execution of user functions, in a manner that required little programming expertise (Duggan, Col. 2, l. 31-46).

Regarding dependent claim 16, while Bobby does not explicitly teach the limitations of claim 16, Duggan teaches inputting criterion to perform logical operations, such as locating certain text in the body of a response from an application program, or verifying that certain text is not present in an expected response (Col. 20, l. 31-35), which would allow the user to select criterion including offensive material.
Both Bobby and Duggan are analogous art because both are testing tools for web applications. It would have been obvious to one of ordinary skill in the art at the time of

the invention to apply Duggan to Bobby, so that Bobby would have the benefit of a simple, easy way to generate and modify a number of test scripts, and verify proper execution of user functions, in a manner that required little programming expertise (Duggan, Col. 2, l. 31-46).

Regarding dependent claim 17, Bobby teaches terminating the analysis in response to a predetermined number of hypertext documents being analyzed (p. 8-9), as specified by link level.

Regarding dependent claim 18, Bobby teaches storing a record of each hypertext document accessed for analysis to prevent previously accessed hypertext documents from being accessed again due to being referenced by another hypertext document (p. 9, "Reanalyzing pages that have changed"). Bobby discloses analyzing stored copies of hypertext documents on a user's computer (Bobby, p. 8, "Entering a URL to test", par. 2).

Regarding dependent claim 19, while Bobby does not explicitly teach the limitations of claim 19, Duggan teaches inputting criterion to perform logical operations, such as locating certain text in the body of a response from an application program, or verifying that certain text is not present in an expected response (Col. 20, l. 31-35), which would allow the user to input a name to be identified for a name change.

Both Bobby and Duggan are analogous art because both are testing tools for web applications. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Duggan to Bobby, so that Bobby would have the benefit of a simple, easy way to generate and modify a number of test scripts, and verify proper

execution of user functions, in a manner that required little programming expertise (Duggan, Col. 2, l. 31-46).

Regarding dependent claim 20, Bobby teaches identifying errors associated with compliance of the hypertext documents with the selected accessibility guidelines; and suggesting a correction for the errors in the summary report (p. 7-8, "Sections of the Report").

Regarding dependent claims 21 and 22, while Bobby in view of Duggan does not explicitly teach suggesting a correction to an indefinable error to a user such that the user may review and approve or disapprove the correction, or prompting a user for a correction such that the user may input the correction, Kondoh teaches a method of testing a hypertext document for compliance with a selected criterion, where the user can select the criterion (col. 3, l. 50-col. 4, l. 15). Kondoh teaches allowing the user to review and approve or disapprove the suggested correction (col. 3, l. 57-col. 4, l. 21). While Kondoh does not explicitly teach prompting a user for a correction such that the user may input the correction, prompting the user for a correction and displaying an error was well known in the art at the time of the invention, since the method was used in spell checking programs and auto completion, for example, at the time of the invention. Bobby, Duggan, and Kondoh are analogous art since all three inventions are directed toward error detection and parsing the structure of web applications. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the selective automatic and user defined error correction of Kondoh with the testing GUI of Duggan and the user defined compliance testing method of Bobby, so

that the result of the error correction would reflect the intent of the document's creator, thereby allowing more flexible and selective error correction (Kondoh, col. 3, l. 50-62).

6. Claims 5-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bobby in view of Nentwich et al. (hereinafter "Nentwich"), "xlinkit: A Consistency Checking and Smart Link Generation Service", *ACM Transactions on Internet Technology*, Vol. 2, No. 2, May 2002, p. 151-185, and further in view of Kondoh.

Independent claim 5 cites: *A method of testing a hypertext document for compliance with a selected criterion, said method comprising: accepting a user selection of the selected criterion and a user input at a client terminal, the user input having at least one parameter indicative of compliance with the selected criterion for an analysis of the hypertext document; transmitting the user selection and the user input to a server, the server having a rules engine component for analyzing the hypertext document based on the user selection and the user input; performing the analysis of the hypertext document at the server and generating analysis data, the analysis data comprising an indication of whether the hypertext document is in compliance with the selected criterion; presenting the analysis data to an application program to correct a definable error in the hypertext document; and presenting at least a portion of the analysis data to the user at the client terminal.* Bobby teaches a method of testing a hypertext document for compliance using the *bobbycl* function, which allows the user to input parameters to test for compliance to a selected criterion such as browser or HTML specification, or accessibility requirement

(p. 12-13, "Using *bobbycl*"). Bobby teaches a dialog which allows a user to select the a specific browser compatibility test (p. 11, "Browser compatibility tests"). Bobby teaches performing an analysis of a hypertext document for compliance and the generation of a summary report of analysis data showing whether a document is in compliance, and storing and presenting the reports (p. 8, "View a report", p. 7-8, "Reading the report). Bobby teaches storing the analysis data and link information for the document (p. 9, Reanalyzing pages that have changed).

While Bobby does not explicitly teach such a client/server network architecture, Nentwich teaches xlinkit, a lightweight application service that checks the consistency of distributed web content using rules. Nentwich teaches a client/server architecture where a user makes a selection of the document set and rule set to be checked, and the check engine is implemented as a servlet on the web server (p. 168-169, "Architecture"). The user may input their own document and rule set. The analysis of rules is performed at the server and the result page is returned to the user at the client browser.

While Bobby allows a user to select from a group of compliance tests, i.e., the selected criterion, Bobby does not explicitly teach user input of a parameter indicative of compliance. However, Nentwich teaches a client/server architecture where a user makes a selection of the document set and rule set to be checked, and the check engine is implemented as a servlet on the web server (p. 168-169, "Architecture"). The user may also input their own document and rule set, i.e., parameters indicative of

compliance. The analysis of rules is performed at the server and the result page is returned to the user at the client browser.

Both Bobby and Nentwich are directed toward web compliance testing using rules. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Nentwich to Bobby, so that the user would have the benefit of a highly generic technology which could be applied to link generation and content management (Nentwich, "Applications", p. 174-175).

While Bobby in view of Nentwich does not explicitly teach presenting the analysis data to an application program to correct a definable error in the hypertext document, Kondoh teaches a method of testing a hypertext document for compliance with a selected criterion, where the user can select the criterion (col. 3, l. 50-col. 4, l. 15). Kondoh teaches correcting a definable error by presenting the analysis data to a program module, i.e., application program (col. 5, l. 13-34). Bobby, Nentwich, and Kondoh are analogous art since all three inventions are directed toward parsing the structure of web applications to determine compliance with rules. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the selective automatic and user defined error correction of Kondoh with the rule based testing and network architecture of Nentwich and the user defined compliance testing method of Bobby, so that the result of the error correction would reflect the intent of the document's creator, thereby allowing more flexible and selective error correction (Kondoh, col. 3, l. 50-62).

Regarding dependent claim 6, Bobby discloses that the user can type in the URL of the page they want to analyze (p. 8, "Entering a URL to test").

Regarding dependent claim 7, Bobby teaches that the user can limit the depth of link levels by a range of zero to Infinite levels (p. 8-9, "Choosing the scope of site analysis").

Regarding dependent claim 8, Bobby discloses a *linkfinder* function which specifies which hypertext links to follow and which documents to analyze (p. 12, "Using *linkfinder*"), and a *bobbycl* function, which allows the user to input parameters to test for compliance to a selected criterion such as browser or HTML specification, or accessibility (p. 12-13, "Using *bobbycl*"). These functions can be run together by the user to generate an analysis of hypertext documents for compliance, and where the analysis data is stored for each document (p. 13-14, "Putting them together"). Bobby teaches presenting compiled summary information about the collection of linked hypertext documents, and identifying what criterion is most problematic, in the form of a list of all issues found, grouped by priority (p. 10, "Summary Reports").

Regarding independent claim 9, claim 9 reflects substantially similar subject matter as claimed in independent claim 5, and in addition, the limitation of *transmitting the analysis data to a server, and storing the analysis data at the server in a non-volatile memory*, which is disclosed by Nentwich. Nentwich teaches that the XML file containing the results of parsing the documents and rule files is stored in the server's local storage before being returned to the user (p. 168-169, "Architecture"). Both Bobby and Nentwich are directed toward web compliance testing using rules. It would have been

obvious to one of ordinary skill in the art at the time of the invention to apply Nentwich to Bobby, so that the user would have the benefit of a highly generic technology which could be applied to link generation and content management (Nentwich, "Applications", p. 174-175). Bobby, Nentwich, and Kondoh are analogous art since all three inventions are directed toward parsing the structure of web applications to determine compliance with rules. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the selective automatic and user defined error correction of Kondoh with the rule based testing and network architecture of Nentwich and the user defined compliance testing method of Bobby, so that the result of the error correction would reflect the intent of the document's creator, thereby allowing more flexible and selective error correction (Kondoh, col. 3, l. 50-62).

Regarding dependent claim 10, Bobby discloses that the user can type in the URL of the page they want to analyze (p. 8, "Entering a URL to test"), compare to *accepting a user designation of the hypertext document to be analyzed.*

Regarding dependent claim 11, Bobby teaches that the user can limit the depth of link levels by a range of zero to Infinite levels (p. 8-9, "Choosing the scope of site analysis"). Compare to *accepting a user designation of the number of link levels from the hypertext document to be analyzed.*

Regarding dependent claim 12, Bobby discloses a *linkfinder* function which *specifies* which hypertext links to follow and which documents to analyze (p. 12, "Using *linkfinder* "), and a *bobbycl* function, which allows the user to input parameters to test for compliance to a selected criterion such as browser or HTML specification, or

accessibility (p. 12-13, "Using *bobbycf*"). These functions can be run together by the user to generate an analysis of hypertext documents for compliance, and where the analysis data is stored for each document (p. 13-14, "Putting them together"). Bobby teaches presenting compiled summary information about the collection of linked hypertext documents, and identifying what criterion is most problematic, in the form of a list of all issues found, grouped by priority (p. 10, "Summary Reports"), therefore it was obvious that the analysis data was stored at the server, so that it could be presented to and accessed by the client browser.

Regarding independent claim 13, claim 13 reflects the server, user interface, and processor components operational for implementing the methods as claimed in claim 5, and is rejected along the same rationale.

Regarding dependent claim 14, while Bobby teaches storing the analysis data in non-volatile memory on the hard drive, Bobby does not explicitly teach that the data is stored at a server. However, Nentwich teaches that the XML file containing the results of parsing the documents and rule files is stored in the server's local storage before being returned to the user (p. 168-169, "Architecture"), compare to the *server...further comprising a non-volatile memory operational to store the analysis data and the link information for the hypertext document.* Both Bobby and Nentwich are directed toward web compliance testing using rules. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Nentwich to Bobby, so that the user would have the benefit of a highly generic technology which could be applied to link generation and content management (Nentwich, "Applications", p. 174-175).

Response to Arguments

Applicant's arguments with respect to claims 1-22 have been considered but are moot in view of the new ground(s) of rejection. The new grounds of rejection includes the addition of the Kondoh patent, which is being relied upon to teach the newly claimed limitation: *correcting a definable error by presenting the analysis data to an application program* (Claim 1).

Conclusion

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Chan et al. U.S. Patent No. 6,799,718 B2 issued October 2004

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Amelia Rutledge whose telephone number is 571-272-7508. The examiner can normally be reached on Monday - Friday 9:30 - 6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AR

H Herndon

Heather R. Herndon
Supervisory Patent Examiner
Technology Center 2100